

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF WISCONSIN

WISCONSIN ALUMNI RESEARCH
FOUNDATION,

Plaintiff,

v.

INTEL CORPORATION,

Defendant.

OPINION AND ORDER

08-cv-78-bbc

Plaintiff Wisconsin Alumni Research Foundation is suing defendant Intel Corporation for allegedly infringing United States Patent No. 5,781,752 (the '752 patent). The '752 patent claims a table-based data speculation approach to a computer's processing of program instructions. In general, the '752 patent was intended to improve the speed with which computer processors could execute program instructions without following the written program order.

Although nine terms were presented for construction, I am convinced from the parties' arguments at the hearing, their pre-hearing briefs, the patent claims, patent specification and prosecution history, that construction of 5 of the terms will resolve the

parties' disputes.

OPINION

When construing claims, the starting point is the so-called intrinsic evidence: the claims themselves, the patent specification and the prosecution history. Teleflex, Inc. v. Ficosa North America Corp., 299 F.3d 1313, 1325 (Fed. Cir. 2002). Examination of the claims' language is where the well established process for claim construction begins. "Claim construction must adhere carefully to the precise language of the claims that the patent [examiner] has allowed." Ardisam, Inc. v. Ameristep, Inc., 336 F. Supp. 2d 867, 879 (W.D. Wis. 2004) (citing Autogiro Co. of America v. United States, 384 F.2d 391, 396 (Ct. Cl. 1967)). The language is given its ordinary meaning as it would be understood by one of ordinary skill in the relevant art, given its context and the other patent claims. Rexnord Corp. v. Laitram Corp., 274 F.3d 1336, 1342 (Fed. Cir. 2001). Moreover, district courts must remain aware that "[t]he patent applicant may not have used words consistent with the dictionary definition because an applicant can act as his or her own lexicographer or may disavow or disclaim aspects of a definition 'by using words or expression of manifest exclusion or restriction, representing a clear disavowal of claim scope.'" Ardisam, 336 F. Supp. 2d at 879-80 (quoting Golight, Inc. v. Wal-Mart Stores, Inc., 355 F.3d 1327, 1331 (Fed. Cir. 2004)).

This initial construction is then considered in light of the specification to determine whether the inventor expressed a different meaning for the language, whether the preferred embodiment is consistent with the initial interpretation and whether the inventor specifically disclaimed certain subject matter. Rexnord, 274 F.3d at 1342-43. The specification contains a written description of the invention that is meant to help explain the invention and possibly define claim terms, Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995), but as a general rule, “limitations from the specification are not to be read into the claims.” Golight, 355 F.3d at 1331. Next, the interpretation is examined for consistency with the patent’s prosecution history and any disclaimers made therein. Rexnord, 274 F.3d at 1343.

Last, a court may consult extrinsic evidence, such as dictionaries, treatises and expert testimony for background information and to “shed useful light on relevant art.” Phillips v. AWH Corp., 415 F.3d 1303, 1317 (Fed. Cir. 2005) (internal citations omitted). In general this type of evidence is less reliable than intrinsic evidence in determining the meaning of claim terms and is “unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of the intrinsic evidence.” Id. at 1318-19.

A. Five Key Disputed Terms in the ‘752 Patent

1. Data speculation circuit

I conclude that the term “data speculation circuit” as used in claim 1 of the ‘752 patent means **a circuit that detects data dependence between load and store instructions and that detects mis-speculation by load instructions.** The dispute between the parties regarding “data speculation circuit” centers on whether the circuit detects data dependence between “load/store pairs,” as defendant contends, or between individual “load and store instructions,” as plaintiff contends. Claim 1 merely states that the data speculation circuit detects “data dependence between instructions.” Col. 14, lns. 38-40. Defendant’s contention that the claim should be construed using “load/store pairs” arises from the specification’s preferred embodiment. Although defendant acknowledges that the law is clear about not importing limitations found in the specification into the patent’s claims, it contends that the specification’s explanation of how to identify load/store pairs that mis-speculate is not merely one example but a description of the invention as a whole. See, e.g., SciMed Life Systems v. Advanced Cardiovascular Systems, Inc., 242 F.3d 1337, 1343 (Fed. Cir. 2001) (limitation from specification could be read into claim when specification made it clear that limitation applied to all embodiments). Defendant is mistaken.

It is true that the specification acknowledges the importance of load/store pairs. See, e.g., col. 14, lns. 15-16 (“The present inventors believe that a relatively limited number of LOAD/STORE pairs will create mis-speculation . . .”). However, there are several reasons

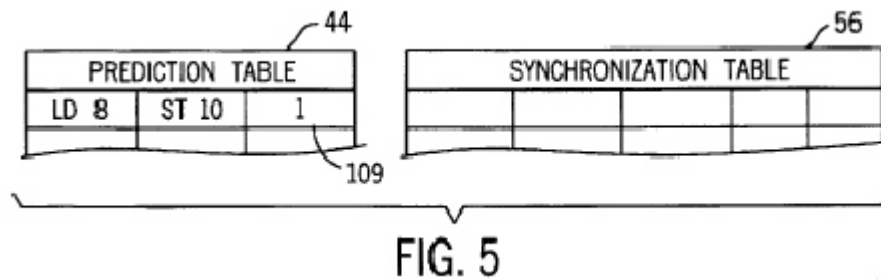
why “load/store pairs” should not be imported into the patent’s claims. First, the inventors clearly understood the importance of load/store pairs, as demonstrated by the discussion of such pairs throughout the specification, see, e.g., col. 3, lns. 53-57; col. 4, ln. 29; col. 14, lns. 15-19, nonetheless, the inventors chose not to use the word “pair” in the claim language. If the inventors had wanted the invention to cover only load/store pairs, they could have worded the claim language accordingly. For example, the inventors could have written the claim to state that a data speculation circuit is used for detecting data dependence between “instruction pairs” as opposed to detecting data dependence between “instructions.” The absence of the word “pair” from the claim language supports the view that the inventors intended the invention to be broader than the specification’s preferred embodiment. Resonate Inc. v. Alteon Websystems, Inc., 338 F.3d 1360, 1367 (Fed. Cir. 2003) (“[L]imitations may not be read into a claim from a preferred embodiment when the claim language is broader than that embodiment.”).

Second, just because the preferred embodiment of the invention is to use load/store pairs to identify mis-speculations does not mean that use of load/store pairs is the only embodiment. Phillips, 415 F.3d at 1332 (“[W]e have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.”). Defendant cites the specification’s embodiment of a “prediction table” as proof that a data speculation circuit requires load/store pairs. Col. 11,

ln. 5. The relevant portion of the specification describes the following embodiment of a prediction table:

[T]he predictor circuit 33 reviews a prediction table 44 shown generally in FIG. 5 to see if the particular instruction 8.2 identified by its physical address is in the prediction table 44.

The prediction table 44 includes three elements for each entry, the elements depicted as columns, the entries as rows. The first column identifies, by physical address, an instruction that is ready for its operation to be preformed; the second column identifies the instruction on which the instruction in the first column may be data dependent; and the third column holds a prediction 109 as will be described.



Col. 11, lns. 4-13. The specification explains that Fig. 5 is “a prediction table 44 shown generally . . .” (Emphasis added). The broad language in the specification indicates that Fig. 5 is merely one embodiment of a prediction table, not the only possible embodiment. Also, the predictor looks at the prediction table for “the particular instruction 8.2 identified by its physical address . . .” (Emphasis added). The focus is on one instruction, not an instruction pair. Although there is a column for a second or paired instruction, this column merely identifies an instruction on “which the instruction in the first column *may* be data

dependent.” Col. 11, lns. 9-10. (Emphasis added). The fact that the load instruction in the first column may or may not be data dependent on more than one store instruction further supports the conclusion that the data speculation circuit should not be limited to focusing on load/store pairs.

The doctrine of claim differentiation provides a third ground for not limiting the invention to load/store pairs. Under that doctrine, “[t]here is presumed to be a difference in meaning and scope when different words or phrases are used in separate claims. To the extent that the absence of such difference in meaning and scope would make a claim superfluous, the doctrine of claim differentiation states the presumption that the difference between claims is significant.” Toro Co. v. White Consolidated Industries, Inc., 199 F.3d 1295, 1302 (Fed. Cir. 1999) (quoting Tandon Corp. v. United States International Trade Commission, 831 F.2d 1017, 1023 (Fed. Cir. 1987)). The data speculation circuit is claimed in independent claim 1 but the prediction table is not introduced until dependent claim 3. Because claim 3 introduces a prediction table, col. 14, lns. 61-62, claim 1 cannot require a prediction table without making claim 3 superfluous. The use of load/store pairs in a prediction table is not evidence that data speculation circuit is limited to load/store pairs.

Any suggestion that the specification clearly limits the invention to only load/store pairs is rebutted by the specification’s discussion of how to use a load instruction alone to

detect data dependence. When the data speculation circuit receives a load instruction and is told to wait to execute the instruction because data speculation is not appropriate, the data speculation circuit waits for one of three signals, col. 10, lns. 35-45, one of which is “a signal indicating that the LOAD instruction is no longer speculative because the earlier STORE instructions did not write to its memory location.” Col. 10, lns. 41-43. (Emphasis added). This section of the specification describes how the data speculation circuit checks the pending load instruction against earlier store instructions, not against one paired store instruction. The specification also discusses checking load instructions, not load/store pairs, when a mis-speculation occurs. See, e.g., col. 13, lns. 5-20 (“[T]he prediction table 44 is again examined but this time for an entry having only the LOAD instruction.”). Therefore, contrary to defendant’s position the specification explains how a data speculation circuit can identify data dependent load and store instructions without using pairs.

Finally, one purpose of the invention is to “identify data dependencies on an on-going or dynamic basis, col. 4, lns. 32-33. This purpose would not be furthered by focusing on only specific load/store pairs. Doing so might overlook mis-speculations; a load instruction could conflict with more than one store instruction. Col. 10, lns. 9-11 (“[I]t is determined . . . whether there are prior STORE instructions on which [the load instruction] might depend.”). As defendant noted at the claims construction hearing, processing data in a computer involves “programs running over and over and over again, millions of times a

second[.] [I]t may be that that same load instruction at some later point in time will need to load something from a different place.” (Tr., Claims Constr. Hrg., dkt. #61, at 74, lns. 6-10.) Thus, programs may be processed in different ways each time they are run. A load instruction dependent on a store instruction may execute before that store instruction during one run of the program but that same dependent load instruction may execute after the same store instruction on a second run of the program, depending on how the instructions are processed. Therefore, if the claim term were construed to cover only load/store pairs, the invention could miss data dependencies that involve the same load instruction but a different store instruction from a previous load/store pair.

In addition to the dispute over load/store pairs, plaintiff requests a construction of data speculation circuit that requires the circuit to track instruction execution. This is discussed in the specification. Col. 7, lns. 4-7. However, the claim does not mention tracking instructions to detect mis-speculations. The claim language is broad enough to cover other potential ways of detecting mis-speculations. Therefore, this limitation from the specification will not be imported into the claim language. Resonate Inc., 338 F.3d at 1367.

2. Mis-speculation

I conclude that the term “mis-speculation” as used in the ‘752 patent means **when a load instruction that is dependent for its data on a store instruction appearing**

earlier in the program order is in fact executed before the store instruction wrote its data to a memory address shared with the load instruction. Claim 1 states that mis-speculation occurs “where a [load] instruction dependent for its data on a [store] instruction of earlier program order, is in fact executed before the [store] instruction.” Col. 14, lns. 41-44. Defendant contends that “mis-speculation” should be construed to mean a “prematurely and erroneously” executed instruction. Although defendant’s contention is not wrong, in the sense that a mis-speculation occurs when a load instruction is executed prematurely and erroneously, col. 7, lns. 40-41, use of the words “prematurely and erroneously” leaves the construction vaguer than the actual claim language; one would have to know what constitutes a premature and erroneous instruction. Construing the term to match the claim language more closely provides a clear explanation of the scope of the term.

3. In fact executed

I conclude that the term “in fact executed” as used in the ‘752 patent means **when a load instruction has actually accessed a memory address that has not yet been updated by a store instruction appearing earlier in the program order.** “In fact executed” occurs in the following context:

where a [load] instruction dependent for its data on a [store] instruction of earlier program order, is in fact executed before the [store] instruction

Col. 14, lns. 40-43. (Underline added). Defendant contends that “in fact executed” means “actually loaded” and plaintiff contends that it includes “or was certain to access.” Neither party is correct. Both the claim language and the specification fail to provide a clear explanation of what “in fact executed” means. The specification explains that a data speculation circuit can keep track of “memory operations as they are *performed* by the processing units so that it can detect any mis-speculations.” Col. 7, lns. 4-7. (Emphasis added). The specification explains further that instructions can be “completely executed” and that “[a]n instruction that has been executed as far as possible is considered ‘ready to commit the operation.’” Col. 7, lns. 20, 24-25. These explanations leave the meaning of “in fact executed” ambiguous because an instruction can be completely executed or “executed as far as possible.”

In claim 1 “in fact executed” refers to a load instruction that is mis-speculated, which means that the instruction was “executed” before a store instruction wrote data to a memory address that the load instruction accessed. Although on its face the meaning of “in fact executed” is somewhat ambiguous, it cannot mean “completed” because an erroneously executed load instruction has not actually been completed. Instead, the mis-speculated load instruction will have to be “squashed,” which means the instruction’s computed value will not be written to memory and the instruction will be reallocated and executed at a later time. Col. 7, lns. 46-48. Therefore, a mis-speculated load instruction attempted to execute, that

is, it executed as far as possible, but reached some error because before a store instruction wrote data to a memory address the load instruction “accessed the same memory address.” Col. 7, lns. 44-45.

In the section of the specification generally discussing processor architecture, the patentees provide some guidance regarding “in fact executed.” The wording in the specification appears to have several typographical errors so the following language is rearranged to provide a comprehensible explanation:

Thus, for example, in the program Table I, if instructions I1 through I3 represent the instruction window **22** and instruction I3 has accessed memory prior to I1 writing to memory [] and at the time I1 is ready to commit [the operation], [the data speculation circuit **30**] checks if I3 has accessed the same memory address as I1 and if so it instruct[s] the allocation circuit **20** and the retirement circuit that I3 is to be squashed and reallocated at a later time.

Col. 7, lns. 48-55. Using as a guide this explanation of the process that occurs once a mis-speculation is detected, “in fact executed” appears to mean “actually accessed” a memory address before the store instruction has written to that memory address. Defendant’s contention that the load instruction must actually load data from a memory address is incorrect in light of the specification’s example, which establishes that it is enough that the load instruction actually *accessed* a memory address for the instruction to have in fact executed.

Plaintiff’s contention that “in fact executed” should be construed to include “or was

certain to access” because a load instruction “is in fact executed as soon as you can determine that it was mis-speculated,” misses the mark as well. (Tr., Claim Constr. Hrg., dkt #61, at 14, lns. 11-13.) Plaintiff contends that a load instruction may take a long time to execute and the processor may begin the execution process before a store instruction has written data to a shared memory address. Then, while the load instruction is executing, but before it actually accesses the data in the shared memory address, a mis-speculation may be detected because the data speculation circuit recognizes that the load instruction was certain to access a memory address shared with the store instruction that has not yet executed. *Id.* at 13, lns. 19-22. Plaintiff’s proposed circumstances are not found in the patent, but are based on general computer science principles that it contends would be known generally to one of ordinary skill in the art. However, even considering what one of ordinary skill in the art would understand generally, the patent’s claims must encompass plaintiff’s suggested embodiment of the invention and the patent’s specification should “teach and enable those of skill in the art to make and use the invention” in that way, Phillips, 415 F.3d at 1323.

Under plaintiff’s proposed circumstances, that is, detecting a mis-speculation before a memory address is actually accessed, “in fact executed” would mean that the load instruction had *begun* to execute prematurely and erroneously or that the load instruction *would* execute prematurely and erroneously. To begin to execute or be executing is not the same as to have “executed.” Plaintiff’s addition of “was certain to access” adds a future

action to what the claim describes as something occurring in the past, and therefore, such a construction is incorrect in light of the claim language. Moreover, the specification's use of the term "accessed" to explain detection of a mis-speculation adds further support to the conclusion that "in fact executed" encompasses the past action of the load instruction actually having accessed the shared memory address and not the future action that the load instruction would eventually access the shared memory address. Therefore, the proper construction of "in fact executed" should not include "or was certain to access" in the future.

4. Predictor

I conclude that the term "predictor" as used in claim 1 of the '752 patent means **a circuit that receives a mis-speculation indication from the data speculation circuit to produce a prediction**. According to claim 1, a predictor produces a "prediction" (a term that will be addressed below). Col. 14, lns. 46-47. Defendant contends that predictor should be construed to require that it produces "a prediction based on historical mis-speculation indications." The specification explains that a predictor does use past mis-speculation indications to update the prediction that it produces. Col. 8, lns. 7-9. However, it also uses the times when an instruction is not mis-speculated to change the prediction value. Col. 12, lns. 14-16 ("[T]he prediction that there was a need to synchronize was wrong and so at process block **120** the prediction **109** is decremented toward the do not

synchronize state.”). Therefore, saying that a predictor operates “based on historical mis-speculation indications” is only partially correct.

Moreover, the claim does not include the phrase “based on historical mis-speculation indications.” Inclusion of such additional language would be importing a limitation from the specification. That is rarely appropriate and particularly not when the limitation is only partially correct. There is no need to add the language when it merely addresses the term “prediction” which will have its own construction. Therefore, any concerns about how a prediction is created can and will be addressed by construing prediction.

5. Prediction

I conclude that the term “prediction” as used in claim 1 the ‘752 patent means **a variable that indicates the likelihood that the data speculative execution of a load instruction will result in a mis-speculation.** Although plaintiff originally asserted an alternate construction, at the claims construction hearing it agreed with this construction. (Tr., Claim Constr. Hrg., dkt #61, at 104, lns. 1-5.) That leaves a dispute similar to the parties’ dispute regarding “data speculation circuit”: whether the prediction indicates the likelihood of mis-speculation by a specific load/store pair or a load instruction. It is unnecessary to discuss the relevant “pair” analysis I applied regarding “data speculation circuit.” There are other pertinent points that support the conclusion to not import

load/store pairs into prediction as used in claim 1.

Beginning with the actual claim language, claim 1 states, “a prediction associated with the particular [load] instruction and based on the mis-speculation indication.” To construe prediction to indicate the likelihood of mis-speculation by only load/store pairs, would require rewriting the claim so that “the particular [load] instruction” was removed from the claim language. It is clear that “[c]ourts do not rewrite claims; instead, [they] give effect to the terms chosen by the patentee.” K-2 Corp. v. Salomon S.A., 191 F.3d 1356, 1364 (Fed. Cir. 1999) (citations omitted). Such rewriting of the claim would be inappropriate.

The specification explains that

the prediction **109** is used to determine the likelihood of a dependency between two instructions in the future. The higher the prediction **109** the more likelihood of mis-speculation if the instruction in the first column is executed before the instruction in the second column.

Col. 14, Ins. 1-6. However, the specification explains further that

It will be understood that the prediction **109** may be obtained by methods other than described herein. . . . More complex pattern matching techniques may be also used, for example, to catch situations where mis-speculations occur in groups or regular patterns.

Col. 14, Ins. 6-14. This section of the specification establishes that linking the prediction to load/store pairs in a prediction table is one way to help limit mis-speculation but that a prediction can be determined using other methods. Therefore, under the claim language and the specification it would be incorrect to limit prediction to load/store pairs.

Moreover, I find plaintiff's three-tier argument persuasive. Plaintiff contends that the patent discloses an invention with three tiers, that claim 1 discloses only the first two tiers and that load/store pairs are not required until tier three. The specification explains a three-tier approach to table-based data dependence speculation. Col. 3, ln. 63 - col. 4, lns. 1-7. Claim 1 focuses on the process in tier two. As explained in the specification, that process is as follows: (1) a load instruction is mis-speculated, as detected by the data speculation circuit; (2) the predictor uses a prediction created from the past mis-speculations for that specific load instruction and determines whether the instruction should be executed or delayed. Col. 3, ln. 67 - col. 4, lns. 1-5. The importance of mis-speculations by specific load/store pairs does not arise until tier three, where a synchronization table is used to determine when a delayed instruction should be performed. Col. 4, lns. 5-7. In the claimed invention, tier three is not disclosed until the "instruction synchronization circuit" is introduced in claim 2. Without the "instruction synchronization circuit," the invention merely predicts whether a load instruction will mis-speculate; when the load instruction has mis-speculated enough, then the "prediction threshold detector" prevents the load instruction from speculating. (Compare claim 1 with claim 2). The "instruction synchronization circuit" does not appear until claim 2 and introduces the concept of instructing "the processor to delay a later execution of the particular [load] instruction until after the particular [store] instruction" Col. 14, lns. 55-57.

Although the specification provides the preferred embodiment of the invention, which includes using load/store pairs in a prediction and synchronization table to predict which specific load/store pairs mis-speculate the most and then delaying the load instruction in that load/store pair until the pair's store instruction executes, the claim language in claim 1 is broader than the preferred embodiment. The law does not require all claims in a patent to cover all embodiments; in fact, some disclosed embodiments may be within the scope of some claims but not others. PSN Illinois, LLC v. Ivoclar Vivadent, Inc., 525 F.3d 1159, 1166 (Fed. Cir. 2008) ("Instead, courts must recognize that disclosed embodiments may be within the scope of other allowed but unasserted claims."). Therefore, it would be improper to limit claim 1 to load/store pairs.

B. Four Remaining Disputed Terms of the '752 Patent

The four remaining disputed terms are:

- (1) where a data consuming instruction dependent for its data on a data producing instruction of earlier program order, is in fact executed before the data producing instruction
- (2) prediction associated with the particular data consuming instruction and based on the mis-speculation indication
- (3) mis-speculation indication
- (4) a prediction threshold detector preventing data speculation for instruction having a prediction within a predetermined range

I conclude that these four terms as used in claim 1 of the '752 patent do not require construction in light of the construction of other claim terms. Any disputes regarding these four terms relates to one of the five claim terms I have construed.

ORDER

IT IS ORDERED that the disputed claim terms of United States Patent No. 5,781,752 are construed as follows:

- “data speculation circuit” as used in claim 1 of the '752 patent means a **circuit that detects data dependence between load and store instructions and that detects mis-speculation by load instructions;**
- “mis-speculation” as used in the '752 patent means **when a load instruction that is dependent for its data on a store instruction appearing earlier in the program order is in fact executed before the store instruction wrote its data to a memory address shared with the load instruction;**
- “in fact executed” as used in the '752 patent means **when a load instruction has actually accessed a memory address that has not yet been updated by a store instruction appearing earlier in the program order;**
- “predictor” as used in claim 1 of the '752 patent means **a circuit that receives a mis-speculation indication from the data speculation circuit**

to produce a prediction;

- “prediction” as used in claim 1 the ‘752 patent means **a variable that indicates the likelihood that the data speculative execution of a load instruction will result in a mis-speculation.**

Entered this 17th day of September, 2008.

BY THE COURT:

/s/

BARBARA B. CRABB
District Judge